IN THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A data structure stored on a computer-readable medium ~~having stored thereon a data structure~~ for use in a computer-aided design and verification system for implementing instrumentation logic in a simulation model of a digital circuit design that includes one or more design entities described utilizing a hardware description language (HDL), said data structure comprising:

~~a first field~~ an assignment statement containing data representing an ~~assigned target~~ instrumentation signal[[;]] and ~~a second field containing data representing~~ a logic or storage element value ~~to be~~ assigned to said ~~assigned target~~ instrumentation signal; and

wherein said assignment statement is incorporated in an HDL source code file using a specified syntax that is recognized by an HDL compiler such that the HDL compiler does not instantiate said instrumentation signal into the digital circuit design, and wherein said specified syntax is subsequently processed by an instrumentation load tool to instantiate said instrumentation signal within a design entity described by said HDL source code file.

2. (Currently Amended) The ~~computer-readable medium~~ data structure of claim 1, wherein said ~~first field and said second~~ assignment statement field ~~are~~ is incorporated within a comment line of ~~a design target entity~~ said HDL source code file and wherein said specified syntax is a non-conventional HDL comment syntax.

3. (Currently Amended) The ~~computer-readable medium~~ data structure of claim 2, ~~further~~ wherein said non-conventional HDL comment syntax is characterized as comprising a pre-pended comment identifier field for identifying said data structure as a ~~hardware description language~~ HDL comment.

4. (Currently Amended) The ~~computer-readable medium~~ data structure of claim 1, wherein said ~~assigned target is~~ instrumentation signal represents a simulation event, and wherein said ~~first~~ assignment statement field further contains data declaring said simulation event.

5. (Currently Amended) The ~~computer-readable-medium~~ data structure of claim 4, wherein said ~~first~~ assignment statement field further comprises:

an event type ~~sub-~~ field containing data ~~for declaring a type of simulation~~ designating an event type for said simulation event; and

an event name field containing data designating an event name for said simulation event.

6. (Currently Amended) The ~~computer-readable-medium~~ data structure of claim 1, wherein said ~~assigned target~~ instrumentation signal is an intermediate instrumentation signal, and wherein said ~~first~~ assignment statement field further contains data declaring said intermediate instrumentation signal.

7. (Currently Amended) The ~~computer-readable-medium~~ data structure of claim 1, wherein said assigned logic value is a logical ~~connectivity~~ expression for combining a plurality of signals within the design entity described by said HDL source code file.

8. (Cancelled)

9. (Cancelled)

10. (New) In a computer-aided design and verification system, a method for implementing instrumentation logic in a simulation model of a digital circuit design that includes one or more design entities described utilizing a hardware description language (HDL), said method comprising:

utilizing a specified syntax to declare instrumentation logic within an HDL source code file, wherein the declaration includes assigning a logic or storage element value to an instrumentation signal; and

compile processing said HDL source code file, wherein said compile processing includes processing said specified syntax to determine whether or not to instantiate said instrumentation signal within a design entity described by said HDL source code file.

11. (New) The method of claim 10, wherein said specified syntax comprises a non-conventional HDL comment syntax.

12. (New) The method of claim 10, wherein said instrumentation signal represents a simulation event.

13. (New) The method of claim 12, wherein said utilizing a specified syntax to declare instrumentation logic comprises:

declaring an event type field containing data designating an event type for said simulation event; and

declaring an event name field containing data designating an event name for said simulation event.

14. (New) The method of claim 10, wherein said assigned logic value is a logical expression for combining a plurality of signals within the design entity described by said HDL source code file.

15. (New) In a computer-aided design and verification system, a system for implementing instrumentation logic in a simulation model of a digital circuit design that includes one or more design entities described utilizing a hardware description language (HDL), said system comprising:

model build processing means for reading an assignment statement containing data representing an instrumentation signal and a logic value assigned to said instrumentation signal, wherein said assignment statement is incorporated in an HDL source code file using a specified syntax; and

compiler means for compile processing said HDL source code file, wherein said compile processing includes processing said specified syntax to determine whether or not to instantiate said instrumentation signal within a design entity described by said HDL source code file.

16. (New) The system of claim 15, wherein said specified syntax comprises a non-conventional HDL comment syntax.

17. (New) The system of claim 15, wherein said instrumentation signal represents a simulation event.

18. (New) In a computer-aided design and verification system, a computer program product for facilitating implementation of instrumentation logic in a simulation model of a digital circuit design that includes one or more design entities described utilizing a hardware description language (HDL), said computer program product including computer-executable instructions for performing a method comprising:

reading an HDL source code file that includes a specified syntax for declaring instrumentation logic within a design entity described in said HDL source code file, wherein the declaration includes assigning a logic or storage element value to an instrumentation signal; and

compile processing said HDL source code file, wherein said compile processing includes processing said specified syntax to determine whether or not to instantiate said instrumentation signal within a design entity described by said HDL source code file.

19. (New) The program product of claim 18, wherein said specified syntax comprises a non-conventional HDL comment syntax.

20. (New) The program product of claim 18, wherein said instrumentation signal represents a simulation event.

21. (New) In a computer-aided design and verification system, a method for implementing instrumentation logic in a simulation model of a digital circuit design that includes one or more design entities described utilizing a hardware description language (HDL), said method comprising:

receiving an HDL source code file that describes a design entity, said HDL source code file including an instrumentation signal declaration having an assignment statement that assigns a logic value to an instrumentation signal, wherein said assignment statement is incorporated in the HDL source code file using a specified syntax; and

responsive to receiving said HDL source code file, compile processing said HDL source code file, wherein said compile processing includes processing said specified syntax to determine whether or not to instantiate said instrumentation signal within the design entity described by said HDL source code file.

22. (New) The method of claim 21, wherein said compile processing comprises:

compile processing said instrumentation signal declaration in a first compile step wherein said specified syntax is processed such that said instrumentation signal is not instantiated within a design entity described by the HDL source code file; and

compile processing said instrumentation signal declaration in a second compile step and in response thereto, instantiating said instrumentation signal within the design entity described by said HDL source code file.

23. (New) The method of claim 22, wherein said first compile step further comprises instantiating the design entity described by the HDL source code file into said simulation model.

24. (New) In a computer-aided design and verification system, a system for implementing instrumentation logic in a simulation model of a digital circuit design that includes one or more design entities described utilizing a hardware description language (HDL), said system comprising:

model build processing means for receiving an HDL source code file that describes a design entity, said HDL source code file including an instrumentation signal declaration having an assignment statement that assigns a logic value to an instrumentation signal, wherein said assignment statement is incorporated in the HDL source code file using a specified syntax; and

model build processing means responsive to receiving said HDL source code file, for compile processing said HDL source code file, wherein said compile processing includes processing said specified syntax to determine whether or not to instantiate said instrumentation signal within the design entity described by said HDL source code file.

25. (New) The system of claim 24, wherein said model build processing means for compile processing further comprises:

first compiler means for compile processing said instrumentation signal declaration in a first compile step wherein said specified syntax is processed such that said instrumentation signal is not instantiated within a design entity described by the HDL source code file; and

second compiler means for compile processing said instrumentation signal declaration in a second compile step and in response thereto, instantiating said instrumentation signal within the design entity described by said HDL source code file.

26. (New) The system of claim 25, wherein said first compiler means further comprises processing means for instantiating the design entity described by the HDL source code file into said simulation model.